DETECTING COMPROMISED BALLOTS

RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application No. 60/270,182 filed February 20, 2001, and is a continuation-in-part of each of U.S. Patent Application No. 09/534,836, filed March 24, 2000; U.S. Patent Application No. 09/535,927, filed March 24, 2000, Now Abandoned; and U.S. Patent Application No. 09/816,869 filed March 24, 2001, Now PATENT # 6950948. Each of these four applications is incorporated by reference in its entirety.

5/19/06
JW

TECHNICAL FIELD

[0002] The present invention is directed to the fields of election automation and cryptographic techniques therefor.

BACKGROUND

[0003] The problems of inaccuracy and inefficiency have long attended conventional, manually-conducted elections. While it has been widely suggested that computers could be used to make elections more accurate and efficient, computers bring with them their own pitfalls. Since electronic data is so easily altered, many electronic voting systems are prone to several types of failures that are far less likely to occur with conventional voting systems.

[0004] One class of such failures relates to the uncertain integrity of the voter's computer, or other computing device. In today's networked computing environment, it is extremely difficult to keep any machine safe from malicious software. Such software is often able to remain hidden on a computer for long periods of time before actually performing a malicious action. In the meantime, it may replicate itself to other computers on the network, or computers that have

C. An Attack on the Previous Protocol

[0028] As noted, malicious software cannot conduct *directed* vote corruption without the corruption being detected and later corrected. However, the basic version of the protocol outlined above in some cases may allow *undirected* vote corruption to go undetected as in the following scenario.

[0029] 1. Instead of submitting (X_i, Y_i) as the voter intends, the voter's computer can submit $(X_i, Y_i h^\gamma)$ for any chosen $\gamma \in Z_q$. This will have the effect of transforming a valid vote into an invalid one. When the computer receives the encrypted vote confirmation, it can follow the protocol to compute $K_i h^{\gamma\beta_i} m^{\beta_i}$.

[0030] 2. Were it to display this value, the voter would notice a problem, since it would not match the confirmation dictionary. However, since the malicious software generated, and knows γ, and also knows h^{β_i} from the encrypted confirmation it received, it can compute $\left(h^{\beta_i}\right)^\gamma = h^{\gamma\beta_i}$. By division, it can then compute right value $K_i m^{\beta_i}$ – i.e., the one to match the voter's confirmation dictionary – and display it, thereby fooling the voter. Of course, invalid votes will be detected at tabulation time, but this will usually be too late for corrective action to be taken. Embodiments of the facility guard against such an undirected attack by employing a *voter validity proof* as discussed in the next section.

D. Counter Attack – Voter Validity Proof

[0031] The validity proof constructed by the voter proves to the vote collection center that, (X_i, Y_i), the encrypted decision (ballot) received from voter V_i, is either an encryption of m_y or an encryption of m_n without revealing any information about which of these values it is. Methods for constructing validity proofs of this type can be found in U.S. Patent Application No. 09/535,927, as well as R. Cramer, I. Damgard, B. Schoenmakers, *Proofs of partial knowledge and simplified design of witness hiding protocols*, Advances in Cryptology, CRYPTO '94, Lecture Notes in Computer Science, pp. 174-187, Springer-Verlag, Berlin, 1994, which is hereby incorporated by reference in its entirety. Thus, the validity

5/19/06 JW

NOW ABANDONED

The following is a detailed example of a Secret Value Confirmation exchange. In order to maximize the clarity of the example, several of the basic parameters used – for example, the number of questions on the ballot, and the size of the cryptographic parameters – are much smaller than those that would be typically used in practice. Also, while aspects of the example exchange are discussed below in a particular order, those skilled in the art will recognize that they may be performed in a variety of other orders.

Some electronic election protocols include additional features, such as:

- voter and authority certificate (public key) information for authentication and audit
- ballot page style parameters
- data encoding standards
- tabulation protocol and parameters

As these features are independent of the Secret Value Confirmation implementation, a detailed description of them is not included in this example.

This example assumes an election protocol that encodes voter responses (answers) as a single ElGamal pair. However, from the description found here, it is a trivial matter to also construct a Secret Value Confirmation exchange for other election protocols using ElGamal encryption for the voted ballot. For example, some embodiments of the facility incorporate the homomorphic election protocol described in U.S. Patent Application No. 09/535,927, now abandoned. In that protocol, a voter response is represented by multiple ElGamal pairs. The confirmation dictionary used in this example is easily modified to either display a concatenation of the respective confirmation strings, or to display a hash of the sequence of them.

5/19/06 JW